



PATENT ABSTRACTS OF JAPAN

(11) Publication number: 11327936 A

(43) Date of publication of application: 30 . 11 . 99

(51) Int. Cl.

G06F 9/46
G06F 13/00

(21) Application number: 10127845

(71) Applicant: CANON INC

(22) Date of filing: 11 . 05 . 98

(72) Inventor: YAMADA TETSUYA

(54) DATA BASE MANAGEMENT METHOD AND
DEVICE THEREFOR AND NETWORK
MANAGEMENT METHOD AND DEVICE
THEREFOR

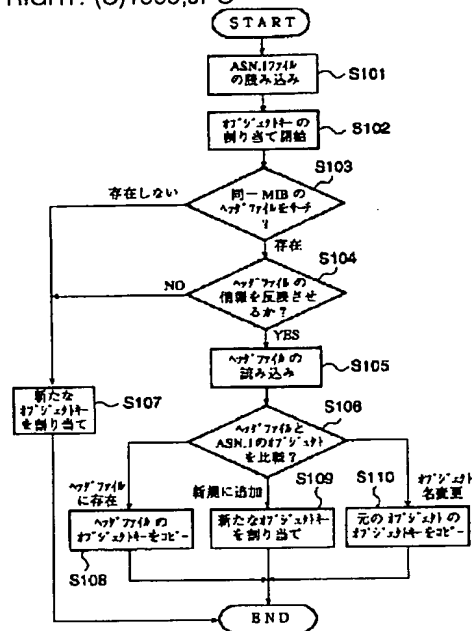
the same index is continuously attached to the
continuously present data.

COPYRIGHT: (C)1999,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To update a network management program just by updating the correspondence table of an object key and an object identifier used in the network management program.

SOLUTION: Whether or not a file for which the allocation of an index is outputted is already present, is searched first (S103), and in the case that an output file is present, the output file is read (S105) and data in the output file and the data under consideration in a data base are compared (S106). As the result of comparison, the index of the pertinent data of the output file is allocated as the index of the data under consideration (S108) in the case that the data under consideration are present in the output file, a new index is allocated to the data under consideration (S109) in the case that the data under consideration are not present in the output file and the index of the desired data of the output file is allocated as the index of the data under consideration (S110) in the case that the name of the data under consideration is changed. In such a manner,



(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 1 1 - 3 2 7 9 3 6

(43) 公開日 平成 11 年 (1999) 11 月 30 日

(51) Int. Cl. °	識別記号	F I
G 0 6 F 9/46	3 6 0	G 0 6 F 9/46 3 6 0 C
13/00	3 5 1	13/00 3 5 1 M

審査請求 未請求 請求項の数 1 1 O L (全 1 4 頁)

(21) 出願番号 特願平 10-127845

(22) 出願日 平成 10 年 (1998) 5 月 11 日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子 3 丁目 30 番 2 号

(72) 発明者 山田 哲也

東京都大田区下丸子 3 丁目 30 番 2 号 キヤノ
ン株式会社内

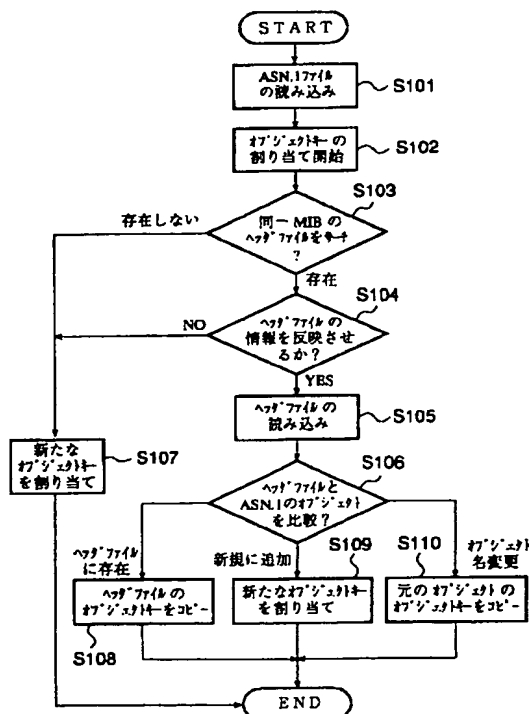
(74) 代理人 弁理士 大塚 康德 (外 2 名)

(54) 【発明の名称】 データベース管理方法及び装置とネットワーク管理方法及び装置

(57) 【要約】

【課題】 ネットワーク管理プログラムで使用しているオブジェクトキーとオブジェクト識別子との対応テーブルを更新するだけでネットワーク管理プログラムを更新する。

【解決手段】 まず、インデックスの割り当てが出力されているファイルが既に存在しているかをサーチし (S 1 0 3)、出力ファイルが存在した場合に、その出力ファイルを読み込み (S 1 0 5)、該出力ファイル中のデータと、前記データベース中の注目データとを比較し (S 1 0 6)、比較の結果、注目データが出力ファイルに存在する場合に、出力ファイルの当該データのインデックスを注目データのインデックスとして割り当て (S 1 0 8)、注目データが出力ファイルに存在しない場合に、注目データに新たなインデックスを割り当て (S 1 0 9)、注目データの名前が変更されている場合に、前記出力ファイルの所望のデータのインデックスを注目データのインデックスとして割り当てる (S 1 1 0)。こうして、継続して存在するデータには、継続して同じインデックスを付す。



【特許請求の範囲】

【請求項 1】 データを、それをノードとするツリー構造で管理するデータベースの各ノードにインデックスを割り当てるデータベース管理方法であって、インデックスの割り当てが出力されているファイルが既に存在しているかをサーチする出力ファイルサーチステップと、

出力ファイルが存在した場合に、その出力ファイルを読み込み、該出力ファイル中のデータと、前記データベース中の注目データとを比較するデータ比較ステップと、比較の結果、前記データベース中の注目データが前記出力ファイルに存在する場合に、出力ファイルの当該データのインデックスを注目データのインデックスとして割り当てるインデックスコピーステップと、比較の結果、前記データベース中の注目データが前記出力ファイルに存在しない場合に、前記注目データに新たなインデックスを割り当てる新規インデックス割り当てステップと、比較の結果、前記データベース中の注目データの名前が変更されている場合に、前記出力ファイルの所望のデータのインデックスを注目データのインデックスとして割り当てるインデックス変換ステップとを有することを特徴とするデータベース管理方法。

【請求項 2】 前記ファイルサーチステップの前に、データベースのデータすべてに新たな異なるインデックスを割り当てるインデックス初期割り当てステップを更に有することを特徴とする請求項 1 に記載のデータベース管理方法。

【請求項 3】 前記新規インデックス割り当てステップは、前記データベース中の注目データが前記出力ファイルに存在せず、かつ、前記出力ファイル中のデータのいずれかから名前が変更されたデータであると指示されなかった場合に、前記注目データに新たなインデックスを割り当て、前記インデックス変換ステップは、前記データベース中の注目データが前記出力ファイルに存在せず、かつ、前記出力ファイル中のデータのいずれかから名前が変更されたデータであると指示された場合に、指示されたデータのインデックスを注目データのインデックスとして割り当てることを特徴とする請求項 1 に記載のデータベース管理方法。

【請求項 4】 前記データベースは、プリンタに関する情報を含むことを特徴とする請求項 1 に記載のデータベース管理方法。

【請求項 5】 データを、それをノードとするツリー構造で管理するデータベースの各ノードにインデックスを割り当てるデータベース管理装置であって、インデックスの割り当てが出力されているファイルが既に存在しているかをサーチする出力ファイルサーチ手段と、

出力ファイルが存在した場合に、その出力ファイルを読

み込み、該出力ファイル中のデータと、前記データベース中の注目データとを比較するデータ比較手段と、比較の結果、前記データベース中の注目データが前記出力ファイルに存在する場合に、出力ファイルの当該データのインデックスを注目データのインデックスとして割り当てるインデックスコピー手段と、

比較の結果、前記データベース中の注目データが前記出力ファイルに存在しない場合に、前記注目データに新たなインデックスを割り当てる新規インデックス割り当て手段と、

比較の結果、前記データベース中の注目データの名前が変更されている場合に、前記出力ファイルの所望のデータのインデックスを注目データのインデックスとして割り当てるインデックス変換手段とを有することを特徴とするデータベース管理装置。

【請求項 6】 データベースのデータすべてに新たな異なるインデックスを割り当てるインデックス初期割り当て手段を更に有することを特徴とする請求項 5 に記載のデータベース管理装置。

【請求項 7】 前記新規インデックス割り当て手段は、前記データベース中の注目データが前記出力ファイルに存在せず、かつ、前記出力ファイル中のデータのいずれかから名前が変更されたデータであると指示されなかった場合に、前記注目データに新たなインデックスを割り当て、前記インデックス変換手段は、前記データベース中の注目データが前記出力ファイルに存在せず、かつ、前記出力ファイル中のデータのいずれかから名前が変更されたデータであると指示された場合に、指示されたデータのインデックスを注目データのインデックスとして割り当てることを特徴とする請求項 5 に記載のデータベース管理装置。

【請求項 8】 前記データベースは、プリンタに関する情報を含むことを特徴とする請求項 5 に記載のデータベース管理装置。

【請求項 9】 請求項 1 に記載のデータベース管理方法により、そのデータとしてネットワークデバイスの情報を管理するネットワーク管理方法。

【請求項 10】 請求項 1 に記載のデータベース管理方法により、そのデータとしてネットワークデバイスの情報を管理するネットワーク管理装置。

【請求項 11】 データベースのデータすべてに新たな異なるインデックスを割り当てるインデックス初期割り当て手段と、インデックスの割り当てが出力されているファイルが既に存在しているかをサーチする出力ファイルサーチ手段と、

出力ファイルが存在した場合に、その出力ファイルを読み込み、該出力ファイル中のデータと、前記データベース中の注目データとを比較するデータ比較手段と、比較の結果、前記データベース中の注目データが前記出力ファイルに存在する場合に、出力ファイルの当該デー

タのインデックスを注目データのインデックスとして割り当てるインデックスコピー手段と、

比較の結果、前記データベース中の注目データが前記出力ファイルに存在しない場合に、前記注目データに新たなインデックスを割り当てる新規インデックス割り当て手段と、

比較の結果、前記データベース中の注目データの名前が変更されている場合に、前記出力ファイルの所望のデータのインデックスを注目データのインデックスとして割り当てるインデックス変換手段とを実現し、データを、それをノードとするツリー構造で管理するデータベースの各ノードにインデックスを割り当てるデータベース管理装置としてコンピュータを機能させるプログラムを格納することを特徴とするコンピュータ可読記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はコンピュータネットワークに関し、特に、ネットワークを管理するネットワーク管理ソフトウェアの開発支援のためのデータベース管理方法及び装置とネットワーク管理方法及び装置に関する。

【0002】

【従来技術】コンピュータネットワークの基板となるコンピュータ機器及びその通信技術の発達により、ここ数年の間に高速で信頼性の高いコンピュータネットワークが出現してきている。それに伴い、ネットワークに接続可能なデバイスの種類が増加し、そのネットワーク管理ソフトウェアが開発されている。このようなネットワーク管理ソフトウェアを使用することによって、ネットワーク上の各種デバイスの情報を取得することができる。

【0003】ネットワーク上のデバイスを管理するための方法としては、これまでに幾つかの試みが数多くの標準期間でなされている。国際標準化機構（ISO）は開放型システム間相互接続（Open System Interconnection, OSI）モデルと呼ばれる汎用基準フレームワークを提供した。ネットワーク管理プロトコルのOSIモデルは、共通管理情報プロトコル（Common Management Information Protocol, CMIP）と呼ばれる。CMIPはヨーロッパの共通ネットワーク管理プロトコルである。

【0004】また米国においては、より共通性の高いネットワーク管理プロトコルとして、簡易ネットワーク簡易プロトコル（Simple Network Management Protocol, SNMP）と呼ばれるCMIPに関連する一変種のプロトコルがある（「シンプルブックインターネット管理入門」

（M. T. ローズ著／松島榮樹訳、（株）トッパン、1995年12月15日発行）を参照）。

【0005】このSNMPネットワーク管理技術によれば、ネットワーク管理システムには、少なくとも1つのネットワーク管理ステーション（NMS）、各々がエー

ジェントを含む幾つかの管理対象ノード、および管理ステーションやエージェントが管理情報を交換するために使用するネットワーク管理プロトコルが含まれる。ユーザは、NMS上でネットワーク管理ソフトウェアを用いて管理対象ノード上のエージェントソフトウェアと通信することにより、ネットワーク上のデータを得、またデータを変更することができる。

【0006】ここでエージェントとは、各々のターゲット装置のバックグラウンドプロセスとして走るソフトウェアである。ユーザがネットワーク上の装置に対して管理データを要求すると、ネットワーク管理ソフトウェアはオブジェクト識別情報を管理パケットまたはフレームに入れてターゲットエージェントへ送り出す。エージェントは、そのオブジェクト識別情報を解釈して、そのオブジェクト識別情報に対応するデータを取り出し、そのデータをパケットに入れてユーザに送り返す。時には、データを取り出すために対応するプロセスが読み出される場合もある。またエージェントは、自分の状態に関するデータをデータベースの形式で保持している。このデータベースのことをMIB（Management Information Base）と呼ぶ。管理する情報の一つ一つはオブジェクトと呼ばれる。図4は、標準として規定されているMIBのうち、一部を抜き出して記載してある。図4に示すように、MIBは木構造のデータ構造をしており、すべてのノードにはオブジェクト名がつけられているとともに、一意に番号付けされている。図4において、括弧内に書かれている番号が、そのノードの識別子である。例えば、図4においてノード401の識別子は1である。ノード402の識別子は、ノード401の下での3なので、1・3と表記される。同様にしてノード403の識別子は、1・3・6・1・2と表記される。このノードの識別子のことをオブジェクト識別子（Object ID）と呼ぶ。

【0007】このMIBの構造は、管理情報構造（SMI：Structure of Management Information）と呼ばれ、RFC1155 Structure and Information and Identification of Management Information for TCP/IP-based Internetsで規定されている。また、SMIは、ASN.1（Abstract Syntax Notation One＝抽出構文記法、1）というOSI言語を使って定義されている。

【0008】ノード404は、SNMPで管理される機器が標準的に備えている標準MIBと呼ばれるオブジェクト群の頂点になるノードであり、このノードの下オブジェクトの詳細な構造については、RFC121 Management Information Base for Network Management of TCP/IP-based internets:MIB-IIに規定されている。

【0009】ノード405は、SNMPで管理されるプリンタが標準的に備えているプリンタMIBと呼ばれるオブジェクト群の頂点になるノードであり、このノード下のオブジェクトの詳細な構造については、RFC1759 Pr

inter MIBで規定されている。

【0010】さらに、ノード406はプライベートMIBと呼ばれ、企業や団体などが独自のMIB定義を行なうための頂点となるノードである。ノード407は企業拡張MIBと呼ばれ、プライベートMIBの中で企業が独自の拡張を行なうための頂点となるノードである。キャノン株式会社には、独自の定義を行うために企業番号として1602が割り当てられており、キャノン独自のMIBであるキャノンMIB(Canon MIB)を定義するための頂点ノード408が、企業を意味するノードであるノード407の下に位置している。キャノンMIBの頂点ノードのオブジェクト識別子は、1・3・6・1・4・1・1602である。

【0011】さて、従来のネットワーク管理ステーションについて以下に述べる。

【0012】ネットワーク管理ステーションは、ASN.1定義からMIBコンパイラと呼ばれるツールを用いて生成されるデータベースを利用してMIBのオブジェクトにアクセスできる。誰でも入手できるMIBコンパイラには、mosy, smicがある。

【0013】ネットワーク管理ステーションにおいて、オブジェクトにアクセスする方法としては、ユーザに個々のオブジェクトを指定させる直接オブジェクトアクセス法と、GUIとしてユーザの親しみやすい情報の形式で提供し、この情報と各オブジェクトの変換モジュールを有する間接オブジェクトアクセス法がある。間接オブジェクトアクセス法を利用したネットワーク管理ステーションの例としては、Hewlett-Packard(HP)社のJetAdminがある。なお、mosy, smicの詳細、入手先等については、前述の「シンプルブック インターネット管理入門」に記載されている。

【0014】

【発明が解決しようとする課題】しかし、上記のようなネットワーク管理ソフトウェアでは、直接オブジェクトアクセス法を用いた場合、オブジェクトの追加、変更、削除などによって管理情報構造に変更が加えられた際には、MIBとユーザとが同じオブジェクトを同じ識別子で指定するために管理情報のデータベースであるMIBのみを更新すれば対処できる利点はあるが、MIBで用いられているオブジェクト識別名などをユーザがあらかじめ知っておく必要があり、操作性が悪いという欠点がある。一方、間接オブジェクトアクセス法を用いた場合、オブジェクトをユーザが親しみ易い名称で指定するため操作性の観点からは優れているが、オブジェクトの追加、変更、削除などによって管理情報構造に変更が加えられた際には、各オブジェクトについてユーザ側の名称とMIB側の識別子との対応関係が変わってしまうことがある。その場合、MIBの情報を利用している制御モジュールなどでは、その対応関係を再構築しなけれ

ばならず、そのために、ネットワーク管理ソフトウェア全体を作成し直すことになり、MIBの変更にとどまらない。

【0015】本発明は上記従来例を鑑みてなされたもので、間接オブジェクトアクセス方式を利用した管理ステーションにおいて、MIBの変更が生じてネットワーク管理ソフトウェアの変更を最小限に抑えるデータベース管理方法及び装置とネットワーク管理方法及び装置を提供することを目的とする。

10 【0016】

【課題を解決するための手段】上記目的を達成するために、本発明は次のような構成からなる。すなわち、データを、それをノードとするツリー構造で管理するデータベースの各ノードにインデックスを割り当てるデータベース管理方法であって、インデックスの割り当てが出力されているファイルが既に存在しているかをサーチする出力ファイルサーチステップと、出力ファイルが存在した場合に、その出力ファイルを読み込み、該出力ファイル中のデータと、前記データベース中の注目データとを比較するデータ比較ステップと、比較の結果、前記データベース中の注目データが前記出力ファイルに存在する場合に、出力ファイルの当該データのインデックスを注目データのインデックスとして割り当てるインデックスコピーステップと、比較の結果、前記データベース中の注目データが前記出力ファイルに存在しない場合に、前記注目データに新たなインデックスを割り当てる新規インデックス割り当てステップと、比較の結果、前記データベース中の注目データの名称が変更されている場合に、前記出力ファイルの所望のデータのインデックスを注目データのインデックスとして割り当てるインデックス変換ステップとを有する。

【0017】あるいは、データを、それをノードとするツリー構造で管理するデータベースの各ノードにインデックスを割り当てるデータベース管理装置であって、インデックスの割り当てが出力されているファイルが既に存在しているかをサーチする出力ファイルサーチ手段と、出力ファイルが存在した場合に、その出力ファイルを読み込み、該出力ファイル中のデータと、前記データベース中の注目データとを比較するデータ比較手段と、比較の結果、前記データベース中の注目データが前記出力ファイルに存在する場合に、出力ファイルの当該データのインデックスを注目データのインデックスとして割り当てるインデックスコピー手段と、比較の結果、前記データベース中の注目データが前記出力ファイルに存在しない場合に、前記注目データに新たなインデックスを割り当てる新規インデックス割り当て手段と、比較の結果、前記データベース中の注目データの名称が変更されている場合に、前記出力ファイルの所望のデータのインデックスを注目データのインデックスとして割り当てるインデックス変換手段とを有する。

【0018】あるいは、上記データベース管理方法により、そのデータとしてネットワークデバイスの情報を管理するネットワーク管理方法。

【0019】あるいは、上記データベース管理方法により、そのデータとしてネットワークデバイスの情報を管理するネットワーク管理装置。

【0020】あるいは、データベースのデータすべてに新たな異なるインデックスを割り当てるインデックス初期割り当て手段と、インデックスの割り当てが出力されているファイルが既に存在しているかをサーチする出力
10 ファイルサーチ手段と、出力ファイルが存在した場合に、その出力ファイルを読み込み、該出力ファイル中のデータと、前記データベース中の注目データとを比較するデータ比較手段と、比較の結果、前記データベース中の注目データが前記出力ファイルに存在する場合に、出力ファイルの当該データのインデックスを注目データのインデックスとして割り当てるインデックスコピー手段と、比較の結果、前記データベース中の注目データが前記出力ファイルに存在しない場合に、前記注目データに
20 新たなインデックスを割り当てる新規インデックス割り当て手段と、比較の結果、前記データベース中の注目データの名前が変更されている場合に、前記出力ファイルの所望のデータのインデックスを注目データのインデックスとして割り当てるインデックス変換手段とを実現し、データを、それをノードとするツリー構造で管理するデータベースの各ノードにインデックスを割り当てるデータベース管理装置としてコンピュータを機能させるプログラムを格納することを特徴とするコンピュータ可読記憶媒体。

【0021】

【発明の実施の形態】本発明の実施の形態として、ネットワーク管理ソフトウェアが実行される装置としてパーソナルコンピュータを用いたシステムについて説明する。

<パーソナルコンピュータ（PC）の構成>図5は、ネットワーク管理プログラムが稼動可能なPCの構成を示すブロック図である。図5において、PC500はネットワーク管理ソフトウェアが稼動するPCであり、例えば図4におけるノード408などに相当する。PC500は、ROM502もしくはハードディスク（HD）5
11に記憶された、あるいはフロッピーディスクドライブ（FD）512より供給されるネットワーク管理プログラムを実行するCPU501を備え、システムバス504に接続される各デバイスを総括的に制御する。

【0022】RAM503は、CPU501の主メモリ、ワークエリア等として機能する。キーボードコントローラ（KBC）505は、キーボード（KB）509や不図示のポインティングデバイス等からの指示入力を制御する。CRTコントローラ（CRTC）506は、CRTディスプレイ（CRT）510の表示を制御す
50

る。ディスクコントローラ（DKC）507は、ブートプログラム、種々のアプリ、編集ファイル、ユーザファイルそしてネットワーク管理プログラム等を記憶するハードディスク（HD）511およびフロッピーディスク
コントローラ（FD）512とのアクセスを制御する。ネットワークインターフェースカード（NIC）508は、LAN100を介して、エージェントあるいはネットワーク機器と双方向にデータをやり取りする。

<ネットワーク管理ソフトウェアのモジュール構成>次に、本発明により生成されるMIBのオブジェクトのデータベースを利用するネットワーク管理ソフトウェアの構成について説明する。

【0023】本発明のネットワーク管理装置は、図5に示したような従来のネットワーク管理装置を実現可能なPCと同様の構成のPC上に実現される。ハードディスク（HD）511には、後述のすべての説明で動作主体となる本発明に係るネットワーク管理ソフトウェアのプログラムが格納される。後述のすべての説明において、特に断りのない限り、実行の主体はハードウェア上はC
20 PU501である。一方、ソフトウェア上の制御の主体は、ハードディスク（HD）511に格納されたネットワーク管理ソフトウェアである。本実施例においては、OSとしては例えば、ウィンドウズ95（マイクロソフト社製）を想定しているが、これに限るものではない。

【0024】なお本願に係るネットワーク管理プログラムは、フロッピーディスクやCD-ROMなどの記憶媒体に格納された形でも良く、その場合には図5に示すフロッピーディスクコントローラ（FD）512または不図示のCD-ROMドライブなどによって記憶媒体から
30 プログラムが読み取られ、ハードディスク（HD）511にインストールされる。

【0025】図6は、本発明に係るネットワーク管理ソフトウェアのモジュール構成図である。

【0026】本発明に係るネットワーク管理ソフトウェアは、図5におけるハードディスク511に格納されており、CPU501によって実行される。その際、CPU501はワークエリアとしてRAM503を使用する。

【0027】図6において、デバイスリストモジュール601は、ネットワークに接続されたデバイスを一覧にして表示するモジュールである。全体制御モジュール602は、デバイスリストからの指示をもとに、他のモジュールを統括する。コンフィグレータ603は、エージェントのネットワーク設定に関する特別な処理を行うモジュールである。探索モジュール604は、ネットワークに接続されているデバイスを探索するモジュールである。探索モジュール604によって探索されたデバイスが、デバイスリスト601によって一覧表示される。NetWareジョブモジュール605は、プリントジョブの状況
50 況をNetW API 616を用いてネットワークサーバーから

取得する。なお、NetWare APIについては、例えばノベル社から発行されている“NetWare Programmer's Guide for C”等に記載されている。UIモジュール606および607は、後述するデバイス詳細ウィンドウを表示するためのモジュールであり、詳細情報を表示する対象機種毎にUIモジュールが存在する。制御モジュール608および609は、詳細情報を取得する対象機種に特有の制御を受け持つモジュールである。UIモジュール606、607と同様に、制御モジュール608、609も詳細情報を表示する対象機種毎に存在する。制御Aモジュール608および制御Bモジュール609は、MIBハンドリングモジュール610を用いて管理対象デバイスからMIBデータを取得し、必要に応じてデータの変換を行い、各々対応するUIAモジュール606またはUIBモジュール607にデータを渡す。

【0028】MIBハンドリングモジュール610は、上位モジュールとSNMPプロトコルモジュール611との間の情報のハンドリングを行う。ここでMIBハンドリングモジュール610の重要な機能として、オブジェクト識別子 \leftrightarrow オブジェクトキー変換がある。オブジェクトキーとは、オブジェクトを指定するインデックスであり、オブジェクト識別子と一対一に対応する32ビットの整数のことである。オブジェクト識別子は可変長の識別子であり、ネットワーク管理ソフトウェアを実装する上で扱いが面倒なので、本発明に係るネットワーク管理ソフトウェアにおいては、オブジェクト識別子と一対一に対応する固定長の識別子であるオブジェクトキーを内部的に用いている。MIBハンドリングモジュール610より上位のモジュールは、このオブジェクトキーを用いてMIBの情報を扱う。これにより、ネットワーク管理ソフトウェアの実装が大幅に削減される。このオブジェクト識別子 \leftrightarrow オブジェクトキー変換はOIMモジュール618を呼び出して行う。OIMモジュールの詳細と各オブジェクトに対してオブジェクトキーを割り当てる方法については後述する。

【0029】SNMPプロトコルモジュール611は、SNMPパケットの送信と受信を行う。

【0030】共通トランスポートモジュール612は、SNMPデータを運搬するための下位プロトコルの差を吸収するモジュールである。実際には、動作時にユーザが選択したプロトコルによって、IPXハンドラ613かUDPハンドラ614のいずれかがデータを転送する役割を担う。なお、UDPハンドラ614は、実装としてWinSock API 617を用いている。なお、WinSockについては、例えばWindows Socket API v1.1の仕様書に記載されている。このドキュメントは、複数箇所から入手可能であるが、例えばマイクロソフト社製のコンパイラであるVisual C++に同梱されている。

【0031】コンフィグレータ603が用いる現在のプロトコル615というのは、動作時にユーザが選択して

いるIPXプロトコルかUDPプロトコルのいずれかのことを示す。

【0032】なお、以下の説明において、本願に係るネットワーク管理ソフトウェアのことを「NetSpot」と呼称する。

<OIMモジュールとMIBMaker>さて、図7にOIMモジュール618の構成例を示す。OIMモジュール618は、1つ以上のOIMテーブル702~704と1つのOIMマネージャ701からなる。OIMテーブル702~704は、オブジェクトキーとオブジェクト識別子の変換のためのテーブルである。OIMテーブルは、各MIBの種別ごとに存在する。例えば図7では、Printer-MIB、HostResources-MIB、Canon-MIBの3種類のMIBのOIMテーブル702~704が存在する。OIMマネージャ701は、管理ソフトウェアの初期化時にすべてのOIMテーブルをロードし、MIBモジュール618の要求に応じて適当なOIMテーブルを参照し、オブジェクト識別子とオブジェクトキーの変換を行う。

【0033】OIMテーブルは、ASN.1ファイルに記述された情報をもとに生成されるものであり、本発明に係るオブジェクト識別子 \leftrightarrow オブジェクトキー変換テーブル作成ツールにより作成されるソースファイルをコンパイルすることにより生成される。なお、以下の説明において、本願に係るオブジェクト識別子 \leftrightarrow オブジェクトキー変換テーブル作成ツールのことを「MIBMaker」と呼称する。また、図5のPC500では、ネットワーク管理ソフトウェア開発支援ツールは稼動可能である。本発明に係るネットワーク管理ソフトウェア開発支援プログラムは、図5に示したようなネットワーク管理ソフトウェアを実行可能なPCと同様の構成のPC上に実現され、しかも両者は同時に存在し得る。

【0034】ハードディスク(HD)511には、後述のすべての説明で動作主体となる本発明に係るネットワーク管理ソフトウェア開発支援ツールのプログラムが格納される。後述のすべての説明において、特に断りのない限り、実行の主体はハードウェア上はCPU501である。一方、ソフトウェア上の制御の主体は、ハードディスク(HD)511に格納されたネットワーク管理ソフトウェア開発支援ツールである。本実施例においては、OSは例えば、ウィンドウズ95(マイクロソフト社製)を想定しているが、これに限るものではない。

【0035】なお本発明に係るネットワーク管理ソフトウェア開発支援プログラムは、フロッピーディスクやCD-ROMなどの記憶媒体に格納された形で供給されても良く、その場合には図5に示すフロッピーディスクコントローラ(FD)512または不図示CD-ROMドライブなどによって記憶媒体からプログラムが読み取られ、ハードディスク(HD)511にインストールされる。

【0036】図8は、ネットワーク管理ソフトウェアの開発支援プログラムのひとつであるMIBMakerにより、ASN. 1ファイルからOIMテーブルが作成されるまでの動作を示すものである。ネットワーク管理プログラムで使用するオブジェクトの識別名(オブジェクトキー)とMIBにおけるオブジェクトとは、作成されたOIMテーブルに従ってMIBハンドリングモジュール610により対応づけられる。

【0037】図において、MIBMaker801は、ASN. 1ファイル802を読み込んでソースプログラムファイルを出力する。本例では、ソースプログラムとしてC言語で記述されたプログラムを想定しているため、MIBMaker801は、ヘッダファイル803とソースファイル804の2つのファイルを出力する。ヘッダファイル803は、ソースファイル802がコンパイルされる時に、ソースファイルに組み込まれ、コンパイルされる。ヘッダファイル803では、各オブジェクト名に対して対応するオブジェクトキーが割りつけられている。一方、ソースファイル804には、変換に必要なデータ型および関数が書かれている。この2つのファイルからOIMテーブル805は生成される。

<MIBMakerの動作>図1にMIBMaker801によるMIBのオブジェクトに対するオブジェクトキーの割り当て方法を示す。MIBMaker起動時の画面は図9に示す。

【0038】S101で、MIBMakerはASN. 1ファイルを読み込み、内部にMIBツリーデータ構造を作成する。読み込み後の画面は図10に示す。画面領域1001に、読み込まれたMIBのツリーが描画されている。画面領域1002には、画面領域1001に表示されたMIBのツリーから選ばれたオブジェクト(図10の例ではオブジェクト1003)に関してASN. 1ファイルで記述されていた内容が表示されている。

【0039】S102で各オブジェクトへのオブジェクトキーの割り当てを始める。まずS103でASN. 1ファイルが存在したディレクトリと同一ディレクトリに同じMIBのヘッダファイルが存在するかどうか探す。このディレクトリとは、ヘッダファイルを作製するために用意された、例えば図5におけるハードディスク511に用意されたものである。なお、作成されたヘッダファイルには、その先頭にMIBモジュール名およびヘッダファイルのバージョン情報が書き込まれており、それをもとに最新のヘッダファイルを選ぶ。ここでMIBモジュール名とは、本実施形態のネットワーク管理システムによる管理対象となっているMIBに付された名称である。

【0040】図11に、S103でヘッダファイルが存在したと判定された場合に表示されるメッセージを示す。ここで、ユーザは、後述の方法でこのヘッダファイルのオブジェクトキーの割り当て方を考慮してオブジェ

クトキーを割り当てるかどうか選択する(S104)。S103でヘッダファイルが存在しない場合およびS104で「ヘッダファイルを反映させない」が選択された場合は、後述の方法で新たにオブジェクトキーを割り当てる。

【0041】さて、S104で「ヘッダファイルを反映させる」を選択した場合、S105においてMIBMakerはS103で見つけたヘッダファイルを読み込む。

【0042】S106では、オブジェクトキーを割り当てようとしているオブジェクトが、

1) ヘッダファイルに既に存在するオブジェクト、2) 新たに追加されたオブジェクト、3) オブジェクト名が変更されているオブジェクト、のいずれであるかを調査する。調査方法の詳細については後述する。

【0043】調査の結果に基づき、1) の場合、S109で、S103で見つけたヘッダファイルに書かれているそのオブジェクトのオブジェクトキーを更新後のオブジェクトキーとして出力する。

【0044】2) の場合、S110において、新たなオブジェクトキーを割り当てる。

【0045】3) の場合、S111において、元の名前のオブジェクトに対して割り当てられていたオブジェクトキーを変更後のオブジェクトキーとして割り当てる。

【0046】オブジェクトキー割り当ての詳細については後述する。

<オブジェクトの属性の調査方法>図1のS106において、出力するオブジェクトが既存、新規、変更のいずれのオブジェクトであるかを調査する方法について以下に説明する。図15は、調査の流れを示したフローチャートであり、図1のS106を詳細に示したものである。

【0047】S1501では、見つかったヘッダファイル中のオブジェクトと、これからオブジェクトキーを割り当てるオブジェクトのリストとを比較する。そしてS1502において、ヘッダファイルには存在するが、これからオブジェクトキーを割り当てるオブジェクトのリストには存在しないオブジェクトを削除されたオブジェクトとして、これからオブジェクトキーを割り当てるオブジェクトのリストには存在するがヘッダファイルには存在しないオブジェクトを追加されたオブジェクトとして検出する。しかしながら、オブジェクト名の変更が行われている場合は削除及び追加と単純に区別できないため、ユーザに指定してもらう必要がある。以下はユーザに変更されたオブジェクトを指定させる作業である。

【0048】S1503では、先ほど検出した、削除されたオブジェクトのリストを表示する。図12は、削除および追加されたオブジェクトが検出された後に表示される画面である。図12において、表の左側の列は削除されたオブジェクトのリストである。右側は、以下で記

述する作業により指定された変更後のオブジェクトを表示し、変更がない場合は「削除」と表示する。

【0049】S1504では、以下の作業の0回以上の繰り返しにより、オブジェクトの名前の変更が完了した場合終了とする。図12では、ユーザが[OK]ボタンを押すことで終了する。一方、名前が変更されるオブジェクトがある場合は、ユーザが削除されたオブジェクトのリストから変更するオブジェクトを選択する。図12では、リストから所望のオブジェクトを選択して[変更]ボタンを押す。

【0050】S1506では、新規追加として検出されたオブジェクトのうち、変更後の名前である可能性のあるオブジェクトのリストを表示する。図13は、図12で欄1201を選択した場合に表示されるリストの例である。

【0051】S1507では、表示されたリストから変更後のオブジェクトを選択する。図14は、図13で欄1301を選択した後に表示される画面の例である。欄1401の右側の項目が、欄1201では削除だったものから、図13で選択された欄1301のオブジェクトに変更されて、変更後のオブジェクトとして表示されている。

【0052】これらの作業により、MIBモジュールに含まれている全てのオブジェクトは、既存のヘッダファイルの内容と比較して、新規、既存、変更のいずれであるかが分かる。

<オブジェクトキーの割り当て>オブジェクトキーの基本的な割り当ては、深さ優先の順で行われる。図2にその例を示す。

【0053】図2において丸囲み数字はオブジェクト識別子である。丸の右下の数字が割り当てられたオブジェクトキーである。例えば、識別子が1.1.1のオブジェクトにはオブジェクトキー1が割り当てられている。実際にMIBに対しての割り当ての場合、MIBのツリーのすべてのノードに対してオブジェクトキーを割り当てるわけではないが、割り当て順はこの方式で割り当てる。すなわち、オブジェクトキーは1から始まる。そして、ツリーを、ルートから枝に沿って走査して、あるノードからその親のノードに戻る際に、そのノードに直前に振った番号の次の番号を振り、これをオブジェクトキーとする。

【0054】S107では、出力するすべてのオブジェクトに対してこの順序でオブジェクトキーを割り当てている。

<MIBの更新時のオブジェクトキーの割り当て>出力するオブジェクトが既存、追加、変更のいずれであるかが前述の方法で分かった後にオブジェクトキーを割り当てる方法について以下で説明する。図3では、MIBの更新時のオブジェクトキーの割り当て例を示している。図3において、丸に網掛けとなっているノードに関して

オブジェクトキーを割り当てる。

【0055】図3において、左側のMIBが右側のMIBに更新されている。図3において、各ノードの丸の中のアルファベットはオブジェクト名を示す。

【0056】左のMIBのオブジェクトEは、右のMIBでは削除されている。また、右のMIBにおいてノードFの下にオブジェクトJおよびKが追加されている。左のMIBのノードIは右のMIBでノードLに対応するが、オブジェクト名が変更されている。また、左のMIBではヘッダファイルの最大オブジェクトキーは6である。

【0057】図3の例を図1のフローに当てはめて説明する。MIBMakerがこれから出力するオブジェクトのリストには、C, D, J, K, H, L, Gの順にオブジェクトが納められている。オブジェクトC, Dは既存のオブジェクトなので、S108で左のMIBのオブジェクトキー1, 2がそれぞれそのまま割り当てられる。続くオブジェクトJは新規のオブジェクトなので、S109で左のMIBの中のオブジェクトキーの最大値 $6 + 1 = 7$ を割り当てる。また次のオブジェクトKも新規オブジェクトなので、S109で先ほどの新規オブジェクトJに割り当てられたオブジェクトキー $7 + 1 = 8$ が割り当てられる。続くオブジェクトHは既存オブジェクトなのでS108でオブジェクトキー4がコピーされる。続くオブジェクトLは、S110で、図15の手順にしたがってオペレータにより左のMIBのオブジェクトIのオブジェクトキー5が割り当てられる。最後のオブジェクトGは既存のオブジェクトなので、S108でオブジェクトキー6がそのまま割り当てられる。

【0058】以上のようにしてMIBMakerによりヘッダファイルを生成すると、ヘッダファイルに含まれるオブジェクト名とオブジェクトキーとの対応表と、別途用意されているオブジェクト名とオブジェクト識別子との対応表とを用いて、オブジェクト識別子とオブジェクトキーとを対応させたOIMテーブルを作成することができる。これにより、ネットワーク管理プログラムがオブジェクトの指定のために用いているオブジェクトキーを、MIBにおけるオブジェクトを指定するためのオブジェクト識別子に一意に対応させることができる。

【0059】すなわち、従来は、MIBのオブジェクトに単純にオブジェクトキーを割り当てていた場合には、MIBの更新が行われた際にあるオブジェクトに対して異なるオブジェクトキーが割り当てられ、既存の制御モジュールがそのオブジェクトに不正なアクセスを行う可能性があったため、その都度ネットワーク管理ソフトウェアを生成しなおさなければならなかった。ところが、本発明により、MIBMakerによりMIBのオブジェクトに対して、構成やオブジェクト名が変更されても継続的にオブジェクトキーを割り当てることができるため、ネットワーク構成やオブジェクト名が変更された場

合でも、変更前からあるオブジェクトに対しては、引き続き同じオブジェクトキーが割り当てられる。その結果、オブジェクトキーによってオブジェクトを指定しているネットワーク管理プログラムを再コンパイルする等して生成しなおす必要がない。

【0060】

【他の実施形態】なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダー、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0061】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読出し実行することによっても達成される。

【0062】この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0063】プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモ리카ード、ROMなどを用いることができる。

【0064】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0065】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0066】

【発明の効果】以上説明したように、本発明によれば、

MIBの更新前に割り当てられていたオブジェクトキーの変更を伴うことなく、更新されたMIBのオブジェクトに対してオブジェクトキーを割り当てることができるようになり、ネットワーク管理プログラムで使用しているオブジェクトキーと、MIBで用いられるオブジェクト識別子との対応テーブルを更新するだけで、ネットワーク管理プログラムを更新できる。

【0067】すなわち、本発明によれば、間接オブジェクトアクセス方式を利用した管理ステーションにおいて、MIBの変更が生じてネットワーク管理ソフトウェアの変更を最小限に抑えることができる。

【0068】

【図面の簡単な説明】

【図1】本発明に係わるオブジェクトキーの割り当て手順のフローチャートである。

【図2】新規にMIBのオブジェクトに対してオブジェクトキーを割り当てる方法の一例の図である。

【図3】MIBの更新時にオブジェクトに対してオブジェクトキーを割り当てる方法の一例の図である。

【図4】MIBの構造を示す概念図である。

【図5】ネットワーク管理ソフトウェアが稼働可能なPCの構成を示すブロック図である。

【図6】ネットワーク管理ソフトウェアのモジュール構成図である。

【図7】OIMモジュールの構成図である。

【図8】OIMテーブルモジュールの作成方法を示すフローチャートである。

【図9】MIBMaker起動時の画面を示す図である。

【図10】MIBMakerがASN.1ファイルを読み込んだ後の画面を示す図である。

【図11】既存のMIBが存在することを知らせるメッセージの例を示す図である。

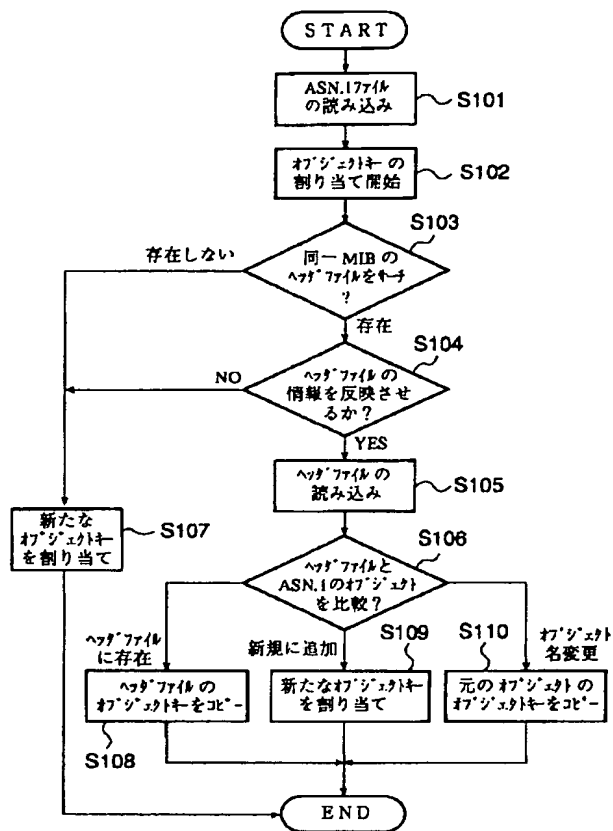
【図12】ヘッダファイルに存在するが出力するファイルにはないオブジェクトのリストを表示した画面の例を示す図である。

【図13】変更後のオブジェクトのリストを表示する画面の例を示す図である。

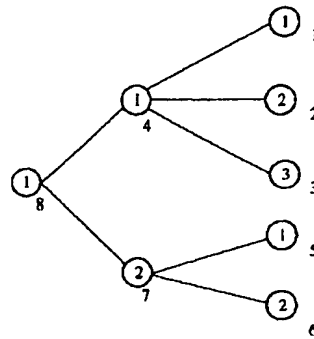
【図14】変更後のオブジェクトを選択した後の画面の例を示す図である。

【図15】出力するオブジェクトが既存、新規追加、変更のいずれであるかを調査する方法のフローチャートである。

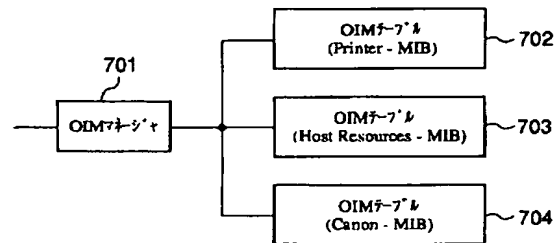
【図 1】



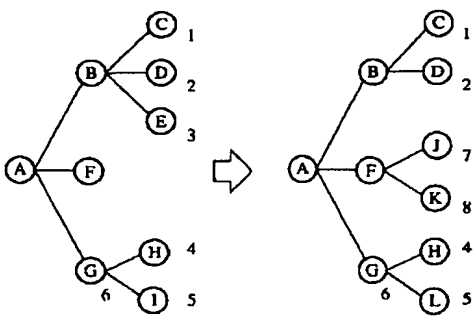
【図 2】



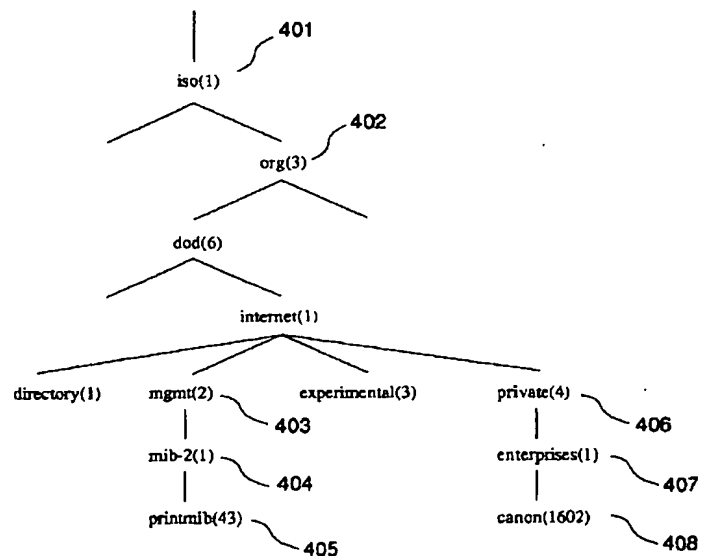
【図 7】



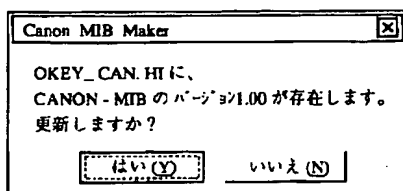
【図 3】



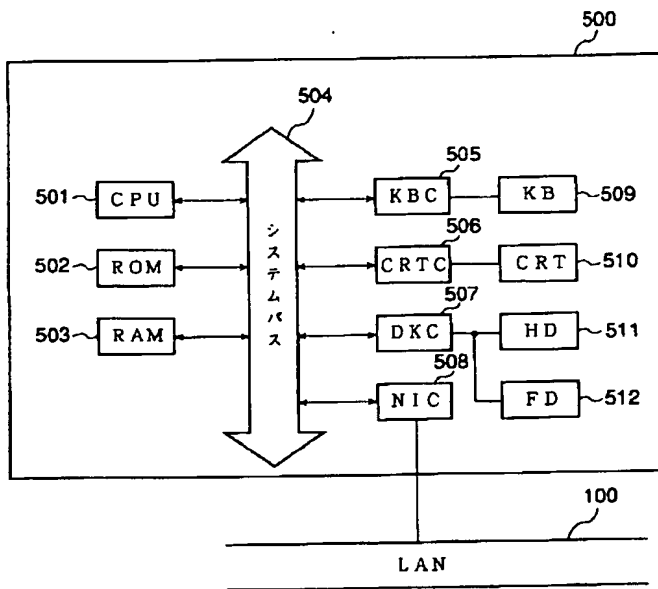
【図 4】



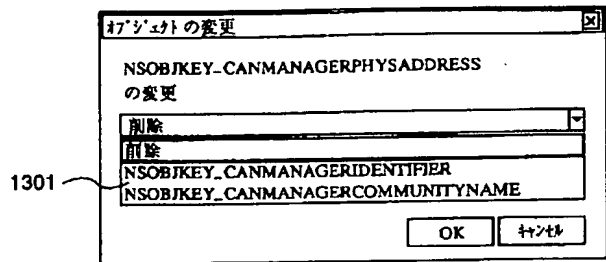
【図 11】



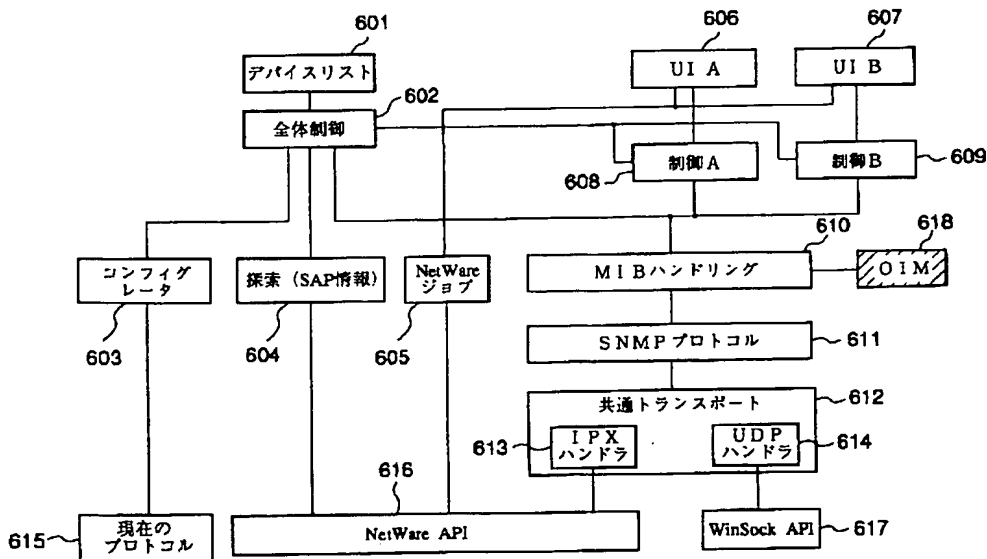
【図 5】



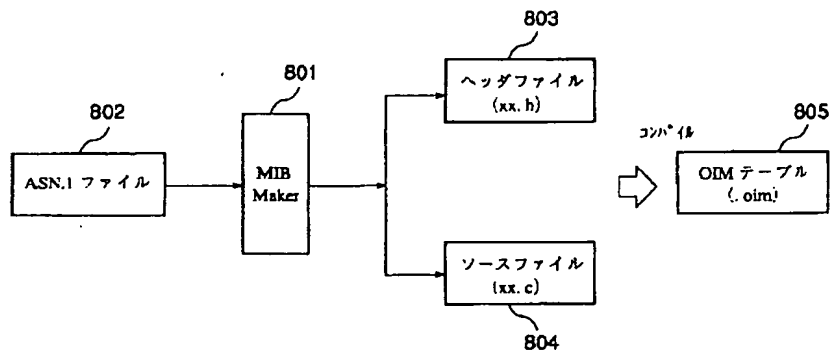
【図 13】



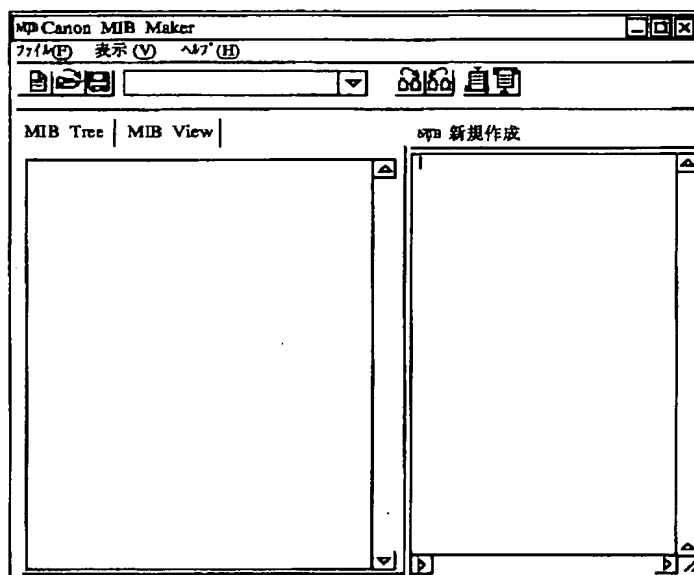
【図 6】



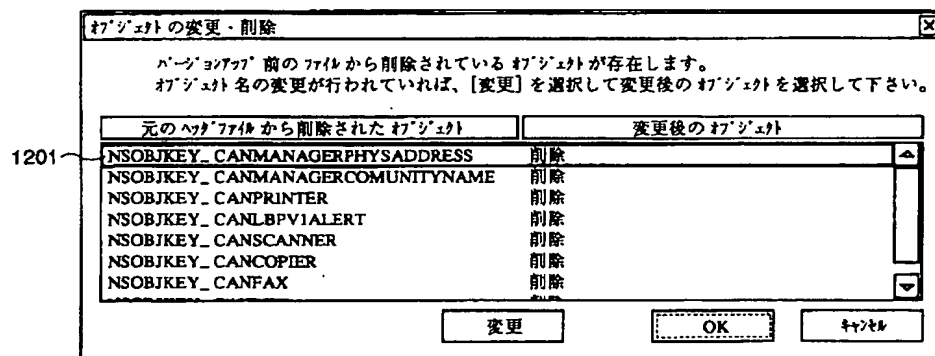
【図 8】



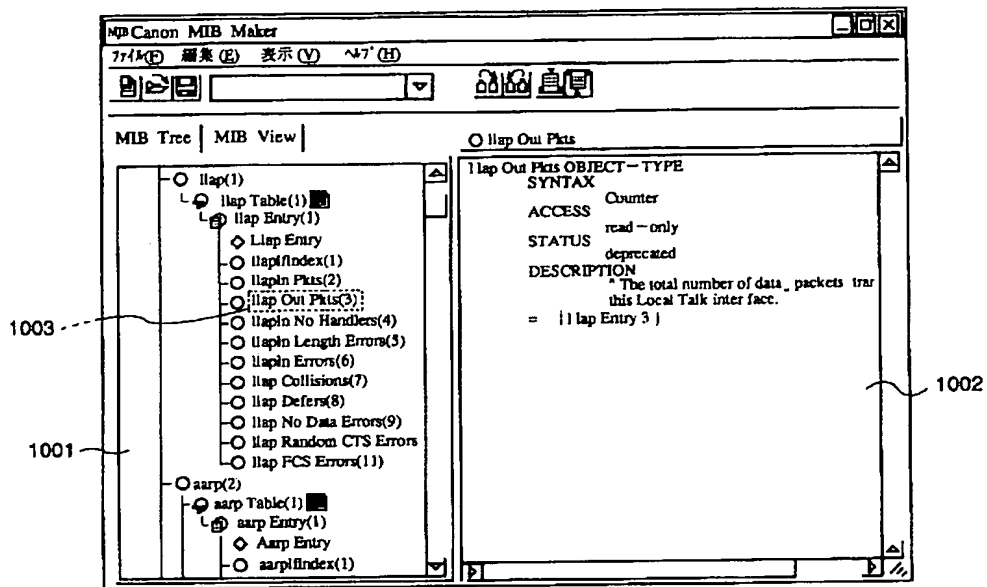
【図 9】



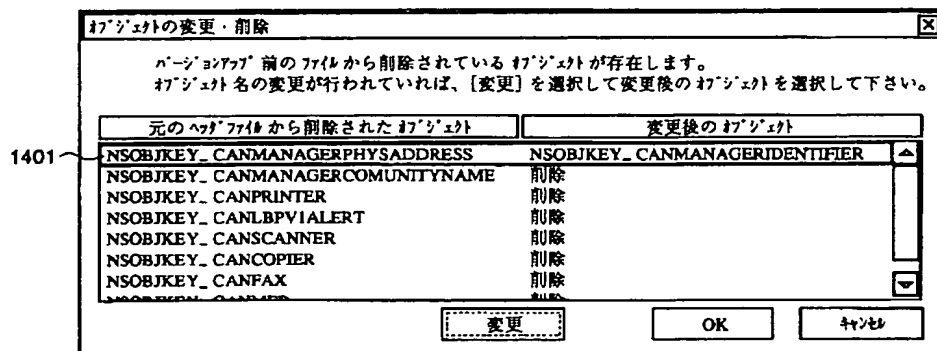
【図 12】



【図 1 0】



【図 1 4】



【図 1 5】

